

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER: _____**

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

This Page Blank (uspto)

(51) Int.Cl.⁶
G 0 6 F 9/46
12/00

識別記号
3 6 0
5 9 1

F I
G 0 6 F 9/46
12/00

3 6 0 C
5 9 1

審査請求 未請求 請求項の数42 FD (全 18 頁)

(21)出願番号 特願平9-293546

(22)出願日 平成9年(1997)10月10日

(31)優先権主張番号 08/729, 421

(32)優先日 1996年10月11日

(33)優先権主張国 米国(US)

(71)出願人 597136607

サンマイクロシステムズ インコーポレーテッド
アメリカ合衆国、94043 カリフォルニア州、マウンテン ビュー、エムエス ピー エーエルアイ-521、ガルシア アヴェニュ 2550

(72)発明者 ウールラス アン エム

アメリカ合衆国、01450 マサチューセツツ州、グラットン、ノースウッズロード9

(74)代理人 弁理士 上野 登

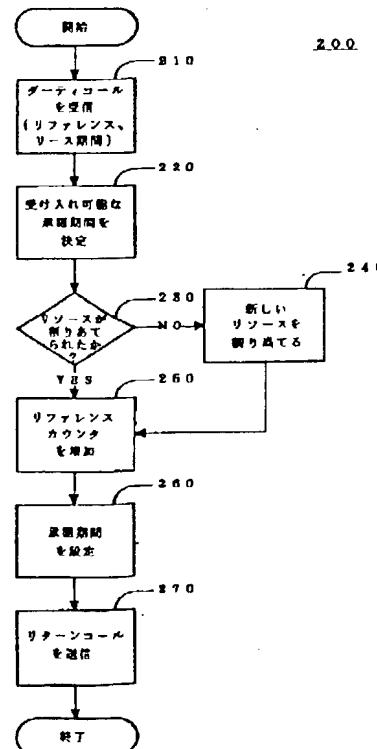
最終頁に続く

(54)【発明の名称】 分散ガーベッジコレクションのためのリソースの管理方法、装置、コンピュータプログラム生産物、コンピュータシステム並びにプラットフォーム間に分散したリソースの割当及び割当解除方

(57)【要約】

【課題】 記憶条件の追加や不必要的通信の増大を招くことなく、リファレンスの保全性が維持され、かつ記憶漏れを起こすことのない分散ガーベッジコレクションのためのリソースの管理方法、装置、コンピュータプログラム生産物、コンピュータシステム並びにリソースの割当及び割当解除方法を提供すること。

【解決手段】 あるシステムのリソースを参照し、かつ要求されたりース期間を指定するリクエスター・プラットフォームからの要求を受信する工程(ステップ210)と、リース期間中はそのシステムリソースへの共有アクセスを許可する工程(ステップ250、260)と、リクエスター・プラットフォームに対してリース期間を通知するリターンコールを送信する工程(ステップ270)と、リース期間が終了した時に、そのシステムリソースの割当を解除する工程とを備えるようにした。



プロセスもアクセスしていないことを示している時に、前記識別されたリソースの割当を解除するためのガーベッジコレクションプロセスを呼び出すサブ工程を備えていることを特徴とする請求項6に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【特許請求の範囲】
【請求項1】 リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信する受信工程と、

承認されたリース期間、前記リソースに対する共有アクセスを許可する許可工程と、

前記プロセスに前記承認されたリース期間を通知する通知工程と、

前記承認されたリース期間が終了した時に前記リソースの割当を解除する割当解除工程とを備えていることを特徴とする分散ガーベッジコレクションのためのリソースの管理方法。

【請求項2】 前記許可工程は、前記要求されたリース期間よりも短い期間となるように前記承認されたリース期間を決定するサブ工程を備えていることを特徴とする請求項1に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【請求項3】 前記許可工程は、前記要求されたリース期間、前記リソースの大きさ、及び前記リソースのための他の承認されたリース期間の少なくとも一つに基づき、前記承認されたリース期間を決定するサブ工程を備えていることを特徴とする請求項1に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【請求項4】 前記許可工程は、前記リソースがそれ以前に割り当てられなかったという決定に基づいて前記リソースを割り当てるサブ工程を備えていることを特徴とする請求項1に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【請求項5】 前記許可工程は、前記リソースに対応するリファレンスカウントを増加させるサブ工程を備え、前記リファレンスカウントは、現在、どれだけ多くのプロセスが前記リソースに対するリファレンスを持っているかを示すものであることを特徴とする請求項1に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【請求項6】 前記プロセスがある識別されたリソースに対するアクセスの終了を望んでいることを示す前記プロセスからの第2の要求を受信する工程と、前記識別されたリソースに対応する前記リファレンスカウントを減少させる工程とをさらに備えたことを特徴とする請求項5に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【請求項7】 前記割当解除工程は、前記リソースに対する共有アクセスがもはや要求されない時に、前記リソースをリクレームするためのガーベッジコレクションプロセスを呼び出すサブ工程を備えていることを特徴とする請求項1に記載される分散ガーベッジコレクションのためのリソースの管理方法。

【請求項8】 前記割当解除工程は、前記リファレンスカウントが前記識別されたリソースに、現在、いかなる

10

20

30

40

50

アクセスすることをプロセスから要求する工程と、前記リソースに対する共有アクセスが許可されている間、承認されたリース期間を前記プロセスから受信する工程と前記承認されたリース期間が終わりかけているが、前記リソースへのアクセスが完了していないという決定に基づいて、新たなリース期間の要求を前記プロセスに対して送信する工程とを備えていることを特徴とする分散ガーベッジコレクションのためのリソースの管理方法。

【請求項10】 前記承認されたリース期間が終了する前に、前記リソースに対する共有アクセスがもはや要求されないことを示す異なる要求を送信する工程をさらに備えていることを特徴とする請求項9に記載される分散ガーベッジコレクションのためのリソースの管理方法

【請求項11】 リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信するように構成された受信モジュールと、承認されたリース期間、前記リソースへの共有アクセスを許可するように構成されたリソースアロケーターと、前記プロセスに前記承認されたリース期間を通知するように構成された通知モジュールと、

前記承認されたリース期間が終了した時に前記リソースの割当を解除するように構成されたリソースディアロケーターとを備えていることを特徴とする分散ガーベッジコレクションのための装置。

【請求項12】 前記リソースアロケーターは、前記要求されたリース期間よりも短い期間となるように前記承認されたリース期間を決定するように構成された決定モジュールを備えていることを特徴とする請求項11に記載される分散ガーベッジコレクションのための装置。

【請求項13】 前記リソースアロケーターは、前記要求されたリース期間、前記リソースの大きさ、及び前記リソースのための他の承認されたリース期間の少なくとも一つに基づき、前記承認されたリース期間を決定するように構成された決定モジュールを備えていることを特徴とする請求項11に記載される分散ガーベッジコレクションのための装置。

【請求項14】 前記リソースアロケーターは、前記リソースがそれ以前に割り当てられなかったという決定に基づいて前記リソースを割り当てるように構成された割当モジュールを備えていることを特徴とする請求項11に記載される分散ガーベッジコレクションのための装置。

【請求項15】 前記リソースアロケーターは、前記リ

ソースに対応するリファレンスカウントを増加させるように構成された増加モジュールを備え、前記リファレンスカウントは、現在、どれだけ多くのプロセスが前記リソースに対するリファレンスを持っているかを示すものであることを特徴とする請求項11に記載される分散ガーベッジコレクションのための装置。

【請求項16】 前記プロセスがある識別されたリソースに対するアクセスの終了を望んでいることを示す前記プロセスからの第2の要求を受信するように構成された第2受信モジュールと、

前記識別されたリソースに対応する前記リファレンスカウントを減少させるように構成された減少モジュールとをさらに備えたことを特徴とする請求項15に記載される分散ガーベッジコレクションのための装置。

【請求項17】 前記リソースディアロケーターは、前記リソースに対する共有アクセスがもはや要求されない時に、前記リソースをリクレームするためのガーベッジコレクションプロセスを呼び出すように構成されたインポーカーを備えていることを特徴とする請求項11に記載される分散ガーベッジコレクションのための装置。

【請求項18】 前記リソースディアロケーターは、前記リファレンスカウントが前記識別されたリソースに、現在、いかなるプロセスもアクセスしていないことを示している時に、前記識別されたリソースの割当を解除するためのガーベッジコレクションプロセスを呼び出すように構成されたインポーカーを備えていることを特徴とする請求項16に記載される分散ガーベッジコレクションのための装置。

【請求項19】 あるリース期間、あるリソースに対してアクセスすることをプロセスから要求するように構成された要求モジュールと、

前記リソースに対する共有アクセスが許可されている間、承認されたリース期間を前記プロセスから受信するように構成された受信モジュールと前記承認されたリース期間が終わりかけているが、前記リソースへのアクセスが完了していないという決定に基づいて、新たなリース期間についての他の要求を、前記プロセスに対して送信するように構成された第2送信モジュールとを備えていることを特徴とする分散ガーベッジコレクションのための装置。

【請求項20】 前記承認されたリース期間が終了する前に、前記リソースに対する共有アクセスがもはや要求されないことを示す異なる要求を送信するように構成された第2送信モジュールをさらに備えていることを特徴とする請求項19に記載される分散ガーベッジコレクションのための装置。

【請求項21】 リソースを管理するために具体化されたコンピュータで読み取り可能なコードを有するコンピュータ使用可能媒体を備えたコンピュータプログラム生産物において、

前記コンピュータ使用可能媒体は、リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信するように構成された受信モジュールと、承認されたリース期間、前記リソースへの共有アクセスを許可するように構成されたリソースアロケーターと、前記承認されたリース期間を通知するように構成された通知モジュールと、

前記承認されたリース期間が終了した時に前記リソースの割当を解除するように構成されたリソースディアロケーターとを備えていることを特徴とする分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項22】 前記リソースアロケーターは、前記要求されたリース期間よりも短い期間となるように前記承認されたリース期間を決定するように構成された決定モジュールを備えていることを特徴とする請求項21に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項23】 前記リソースアロケーターは、前記要求されたリース期間、前記リソースの大きさ、及び前記リソースのための他の承認されたリース期間の少なくとも一つに基づき前記承認されたリース期間を決定するように構成された決定モジュールを備えていることを特徴とする請求項21に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項24】 前記リソースアロケーターは、前記リソースがそれ以前に割り当てられなかつたという決定に基づいて前記リソースを割り当てるように構成された割当モジュールを備えていることを特徴とする請求項21に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項25】 前記リソースアロケーターは、前記リソースに対するリファレンスを現在有しているプロセスの数を示すように構成されたリファレンスカウントモジュールを備えていることを特徴とする請求項21に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項26】 前記コンピュータ使用可能媒体は、前記プロセスがある識別されたリソースに対するアクセスの終了を望んでいることを示す前記プロセスからの第2の要求を受信するように構成された第2受信モジュールと、

前記識別されたリソースに対応する前記リファレンスカウントを減少させるように構成された減少モジュールとをさらに備えていることを特徴とする請求項25に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項27】 前記リソースディアロケーターは、前記リソースに対する共有アクセスがもはや要求されない時に、前記リソースをリクレームするためのガーベッジ

コレクションプロセスを呼び出すように構成されたインポーターを備えていることを特徴とする請求項21に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項28】 前記リソースディアロケーターは、前記リファレンスカウントが前記識別されたリソースに、現在、いかなるプロセスもアクセスしていないことを示している時に、前記識別されたリソースの割当を解除するためのガーベッジコレクションプロセスを呼び出すように構成されたインポーターを備えていることを特徴とする請求項26に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項29】 リソースを管理するために具体化されたコンピュータで読み取り可能なコードを有するコンピュータ使用可能媒体を備えたコンピュータプログラム生産物において、

前記コンピュータ使用可能媒体は、あるリース期間、あるリソースに対してアクセスすることをプロセスから要求するように構成された要求モジュールと、

前記プロセスが前記リソースに対する共有アクセスを許可している間、承認されたリース期間を前記プロセスから受信するように構成された受信モジュールと、

前記承認されたリース期間が終わりかけているという決定に基づいて、新たなリース期間についての他の要求を、前記プロセスに対して送信するように構成された送信モジュールとを備えていることを特徴とする分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項30】 前記コンピュータ使用可能媒体は、前記承認されたリース期間が終了する前に、前記リソースに対する共有アクセスがもはや要求されないことを示す異なる要求を送信するように構成された第2送信モジュールをさらに備えていることを特徴とする請求項29に記載される分散ガーベッジコレクションのためのコンピュータプログラム生産物。

【請求項31】 コンピュータプロセッサーと、前記コンピュータプロセッサーと有効に接続されているメモリーと、

前記メモリーにより前記コンピュータプロセッサー中で実行するコンピュータプロセスを備え、

前記コンピュータプロセスは、リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信するように構成された受信モジュールと、

承認されたリース期間、前記リソースに対する共有アクセスを許可するように構成されたリソースアロケーターと、

前記承認されたリース期間を通知するように構成された通知モジュールと、

前記承認されたリース期間が終了した時に前記リソースの割当を解除するように構成されたリソースディアロケ

ーターとを備えていることを特徴とする分散ガーベッジコレクションのためのコンピュータシステム。

【請求項32】 前記リソースアロケーターは、前記要求されたリース期間よりも短い期間となるように前記承認されたリース期間を決定するように構成された決定モジュールを備えていることを特徴とする請求項31に記載される分散ガーベッジコレクションのためのコンピュータシステム。

【請求項33】 前記リソースアロケーターは、前記要求されたリース期間、前記リソースの大きさ、及び前記リソースのための他の承認されたリース期間の少なくとも一つに基づき、前記承認されたリース期間を決定するように構成された決定モジュールを備えていることを特徴とする請求項31に記載される分散ガーベッジコレクションのためのコンピュータシステム。

【請求項34】 前記リソースアロケーターは、前記リソースがそれ以前に割り当てられなかつたという決定に基づいて前記リソースを割り当てるように構成された割当モジュールを備えていることを特徴とする請求項31に記載される分散ガーベッジコレクションのためのコンピュータシステム。

【請求項35】 前記リソースアロケーターは、前記リソースに対するリファレンスを現在有しているプロセスの数を示すように構成されたリファレンスカウントモジュールを備えていることを特徴とする請求項31に記載される分散ガーベッジコレクションのためのコンピュータシステム。

【請求項36】 前記コンピュータプロセスは、前記プロセスがある識別されたリソースに対するアクセスの終了を望んでいることを示す前記プロセスからの第2の要求を受信するように構成された第2受信モジュールと、前記識別されたリソースに対応する前記リファレンスカウントを減少させるように構成された減少モジュールとをさらに備えていることを特徴とする請求項35に記載される分散ガーベッジコレクションのためのコンピュータシステム。

【請求項37】 前記リソースディアロケーターは、前記リソースに対する共有アクセスがもはや要求されない時に、前記リソースをリクレームするためのガーベッジコレクションプロセスを呼び出すように構成されたインポーターを備えていることを特徴とする請求項31に記載される分散ガーベッジコレクションのためのコンピュータシステム。

【請求項38】 前記リソースディアロケーターは、前記リファレンスカウントが前記識別されたリソースに、現在、いかなるプロセスもアクセスしていないことを示している時に、前記識別されたリソースの割当を解除するためのガーベッジコレクションプロセスを呼び出すように構成されたインポーターを備えていることを特徴とする請求項36に記載される分散ガーベッジコレクショ

ンのためのコンピュータシステム。

【請求項39】 コンピュータプロセッサーと、
前記コンピュータプロセッサーと有効に接続されている
メモリーと、
前記メモリーにより前記コンピュータプロセッサー中で
実行するコンピュータプロセスを備え、
前記コンピュータプロセスは、あるリース期間、あるリ
ソースに対してアクセスすることをプロセスから要求す
るように構成された要求モジュールと、
前記プロセスが前記リソースに対する共有アクセスを許
可している間、承認されたリース期間を前記プロセスか
ら受信するように構成された受信モジュールと、
前記承認されたリース期間が終わりかけているという決
定に基づいて、新たなリース期間についての他の要求
を、前記プロセスに対して送信するように構成された送
信モジュールとを備えていることを特徴とする分散ガ
ーベッジコレクションのためのコンピュータシステム。

【請求項40】 前記コンピュータプロセスは、前記承
認されたリース期間が終了する前に、前記リソースに対
する共有アクセスがもはや要求されないことを示す異な
る要求を送信するように構成された第2送信モジュール
をさらに備えていることを特徴とする請求項39に記載
される分散ガーベッジコレクションのためのコンピュー
タシステム。

【請求項41】 複数のプラットフォームを有する分散
処理システムにおいて、サーバープラットフォームで実
行される以下の工程、すなわち、(a) システムリソ
ースに対するリファレンス及び要求されたリース期間を
含むリクエスター プラットフォームからの要求であつ
て、

前記対応するリクエスター プラットフォームが前記要求
中において前記要求された期間、前記リファレンスされ
たシステムリソースへのアクセスをシークすることを示
している要求を受信する工程と、(b) (i) 前記
要求された各リース期間に応答して設定される、対応する
承認期間であつて、要求に関連する前記リファレンス
された各システムリソースが前記プラットフォームによ
つてアクセスされることが可能となる期間を示して
いる承認期間を設定すること、及び(ii) 対応するリフ
アレンスカウンターであつて、要求の受信によって決定
される前記対応するシステムリソースがアクセスして
いる前記プラットフォームの数を示しているリファレンス
カウンターを増加させること、により前記リファレンス
された各システムリソースを割り当てる工程と、(c)

関連する要求に応答して、関連する要求中でリファレン
スされた前記リソースのための前記承認期間を指定する
前記リクエスター プラットフォームに対する応答を送
信する工程と、(d) (i) リファレンスされた各
システムリソースのための前記承認期間が終了した時、
又は(ii) 前記対応するリファレンスカウンター

が、いかなるリクエスター プラットフォームも、現在、
前記リファレンスされた各システムリソースに対するア
クセスをシークしていないということを示す状態である
時に、前記リファレンスされた各システムリソースの割
当を解除するためのガーベッジコレクションプロセスを
起動する工程とを備えたことを特徴とする分散ガーベッ
ジコレクションのためのプラットフォーム間に分散したリ
ソースの割当及び割当解除方法。

10 【請求項42】 (e) 前記システムリソースに対する
リファレンスを含み、かつ前記リクエスター プラット
フォームが前記システムリソースに対するアクセスの終
了を要求することを示している前記リクエスター プラット
フォームからの新しい要求を受信する工程と、(f)

前記新しい要求において指定された前記リファレンス
された各システムリソースに対応する前記リファレンス
カウンターを減少させる工程とをさらに備えていること
を特徴とする請求項41に記載される分散ガーベッジコ
レクションのためのプラットフォーム間に分散したリソ
ースの割当及び割当解除方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、一般に、コンピュー
ーターシステムのためのガーベッジコレクションに関
し、更に詳しくは、リファレンスと結合し、あるいは関
連しているリソースを収集するための耐故障性の分散ガ
ーベッジコレクションの方法に関する。

【0002】

【従来の技術】 適切なリソース管理は、コンピュータの
効率的及び効果的使用のための重要な観点である。一般
30 に、リソース管理には、適切な時、例えば、要求者がも
はやリソースを要求しない時にリソースの割り当てを解
除することのみならず、要求に応じてリソース（例え
ば、メモリー）を割り当てることも含まれる。一般に、
リソースには、コンピュータ内で実行する演算の構成要
素（例えば、アプリケーション、プログラム、アプレット
など）によって引用されるデータが含まれる。

【0003】 実際には、コンピュータ上で実行している
アプリケーションがリソースを参照するためにシークす
る時には、コンピュータは、アプリケーションが適切に
それらを参照することができるよう、最初にリソース
を割り当てるか、あるいは指定しなければならない。ア
プリケーションが、もはやリソースを参照しないとき
は、コンピュータは、再使用のためにそのリソースの割
り当てを解除するか、あるいはリクームすることができる。
コンピュータ内では、各リソースは、それによ
つてそのリソースが引用可能な唯一の「ハンドル」を持
っている。そのハンドルは、アドレス、配列インデック
ス、特有の値、ポインター等のような、様々な方法によ
って与えられる。

40 50 【0004】 リソース管理は、1台のコンピュータにと

っては比較的単純である。なぜならば、アプリケーションがもはやそれらを参照しない時、あるいは停電のように、リソースがリクレームされることができる時を示す事象は、決定することが容易だからである。多数のコンピュータを接続した分散システムのためのリソース管理は、数台の異なるコンピュータ内のアプリケーションが同一のリソースを使っていることもあるので、より困難である。

【0005】分散システムの分断は、不適切かつ先走ったリソースのリクレーム、あるいはリソースのリクレームの失敗の原因となる。例えば、分散システムにおいて異なるコンピュータ上で実行されている多数のアプリケーションは、他の機械上に配置されているリソースを参照するかもしれない。もし、リソースが配置されているコンピュータと、それらのリソースを参照しているアプリケーションとの間の接続が途切れるならば、コンピュータは、早まってそのリソースをリクレームすることもあり得る。代わりに、コンピュータは、アプリケーションがそのリソースにアクセスすることに失敗してから長期間経過しているにもかかわらず、永続的にそのリソースを保持することもありうる。

【0006】これらの問題は、ネットワークリソースを管理するためのシステムの発展をもたらし、その一つは、「分散ガーベッジコレクション」として知られている。この用語は、ネットワーク中の異なるコンピュータ上で実行されている一つのアプリケーションあるいはアプリケーション群によって用いられているリソースを自動的に管理する、分散システムのための言語又は実行時間システムによって与えられる機能を表している。

【0007】一般に、ガーベッジコレクションは、リソースがもはやアプリケーションのいかなる部分によっても引用されない時に、リソースは、将来の使用のために開放されるという概念を用いている。分散ガーベッジコレクションは、いかなるコンピュータ上のアプリケーションもリソースを参照しない時にリソースをリクレームすることにより、この概念を、分散コンピューティングの領域に拡張する。

【0008】分散ガーベッジコレクションは、割り当てられたリソースとそれらのリソースのリファレンスとの間の保全性を保持しなければならない。換言すれば、そのシステムは、ネットワーク中のいずれかのコンピュータ上で実行されているアプリケーションがそのリソースの参照を続けている時には、リソースの割り当てを解除したり、あるいはリソースを開放することが許されない。「リファレンスの保全性」と称するこのリファレンスとリソースの結合は、リファレンスが参照するリソースへのアクセスを常に承認するということを保証するものではない。例えば、ネットワークの故障は、そのようなアクセスを不可能にする。しかしながら、保全性は、もし、そのリファレンスがいずれかのリソースへの

10 アクセスを得るために用いられたならば、そのリソースは、そのリファレンスが最初に与えられたリソースと同一のリソースであるということを保証するものである。

【0009】ガーベッジコレクションが用いられる分散システムは、また、有限の未来におけるある時点においてもはや引用されなくなったリソースをリクレームしなければならない。換言すれば、そのシステムは、「記憶漏れ」に対する保証を与えるなければならない。記憶漏れは、全てのアプリケーションがリソースに対するリファレンスを落としたが、例えば、いくつかのアプリケーションが未だそのリソースを参照しているという誤った決定をしたために、システムが、再使用のためにリソースをリクレームすることに失敗した時に生ずる。

【0010】リファレンスの保全性の故障と記憶漏れは、しばしば、リソースを引用しているアプリケーションとそれらのリソースの割当及び割当解除を管理しているガーベッジコレクションシステムとの間の分断の原因となる。例えば、リソースを参照しているアプリケーションとそのリソースを管理しているガーベッジコレクションシステムとの間のネットワーク結合の分断は、ガーベッジコレクションシステムが、リソースのリクレームをすべきか否か及びリソースをいつリクレームすべきかを決定することを妨げるかもしれない。代わりとして、ガーベッジコレクションシステムは、アプリケーションが予め定められた時間内にリソースにアクセスしなかつたので、そのリソースを収集してもよいということを誤って決定するかもしれない。

【0011】分散ガーベッジコレクションのメカニズムが記憶漏れを起こすことなくリファレンスの保全性を保持するということを保証することを試みることにより、その分散ガーベッジコレクションのメカニズムを改善するために、多くのテクニックが用いられている。一つのありふれたアプローチは、リファレンスカウントの形式を用いている。そこでは、カウントは、各リソースを参照しているアプリケーションの数を保持している。リソースのカウントがゼロになるとき、ガーベッジコレクションシステムは、そのリソースをリクレームすることができる。

【0012】しかしながら、そのようなリファレンスカウントの機構は、リソースが対応するリファレンスカウンターと共に作られる時にのみ動作する。この場合、ガーベッジコレクションシステムは、追加のアプリケーションがそのリソースを参照するときには、そのリソースのリファレンスカウントを増加させ、あるアプリケーションがもはやそのリソースを参照しない時には、カウントを減少させる。

【0013】しかしながら、リファレンスカウントの機構は、特に、分散システムにおいて故障を生ずるという問題に直面している。そのような故障は、リソースはもはや引用されないということをガーベッジコレクション

システムに知らせるメッセージの伝達を妨げる、コンピュータもしくはアプリケーションの故障、又はネットワークの故障という形をとる。もし、メッセージがネットワークが分断しているために伝達されないならば、不要情報収集システムは、いつリソースをリクームすべきかわからない。

【0014】 そのような故障を避けるために、いくつかのありふれたリファレンスカウントの機構には、「キープアライブ」メッセージが含まれている。これは、また、「ピンバック」とも称されている。この機構によれば、ネットワーク中のアプリケーションは、リソースを監視しているガーベッジコレクションシステムに対してメッセージを送り、そのアプリケーションがなお通信可能であることを示す。これらのメッセージは、ガーベッジコレクションシステムがリソースに対するリファレンスを落とすのを妨げる。

【0015】 そのような「キープアライブ」メッセージの受信の失敗は、ガーベッジコレクションシステムが、あるリソースのためのリファレンスカウントを減少させることができること、そしてすなわち、そのカウントがゼロになるとときは、ガーベッジコレクションシステムは、そのリソースをリクームするかもしれないということを意味している。しかしながら、これは、ネットワークが故障しているために、「キープアライブ」メッセージを受け取ることに失敗したことから、リファレンスカウントがゼロへ到達し、それに続いて、リソースの早まつたリクームが生ずる。これは、リファレンスの保全性の必要条件を乱すものである。

【0016】 ガーベッジコレクションシステムにおけるリファレンスの保全性の問題を解決するために提案された他の方法は、リファレンスカウントのみならず、リソースを参照するコンピュータの各構成要素に対応する識別子もまた保持することである。A. B i r r e l l 他、「ネットワークの目的物のための分散ガーベッジコレクション」、No. 116、ディジタル システムス

リサーチ センター、1993年12月15日参照。

【0017】 この方法は、リファレンスカウントの機構と同一の問題に悩まされている。さらに、この方法は、分散形システム内で不必要的通信を増加させ、かつ記憶のための必要条件（すなわち、各リソースを参照するアプリケーションに対応する識別子のリスト）を追加するオーバーヘッドを付加した唯一の識別子を、各リソースを参照するコンピュータの各構成要素に与えることを必要とする。

【0018】

【発明が解決しようとする課題】 本発明が解決しようとする課題は、記憶条件の追加や不必要的通信の増大を招くことなく、リファレンスの保全性が維持され、かつ記憶漏れを起こすことのない分散ガーベッジコレクションのためのリソースの管理方法、装置、コンピュータプロ

グラム生産物、コンピュータシステム並びにプラットフォーム間に分散したリソースの割当及び割当解除方法を提供することにある。

【0019】

【課題を解決するための手段】 上記課題を解決するために、本発明に係る分散ガーベッジコレクションのためのリソースの管理方法は、リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信する受信工程と、承認されたリース期間、前記リソースに10 対する共有アクセスを許可する許可工程と、前記プロセスに前記承認されたリース期間を通知する通知工程と、前記承認されたリース期間が終了した時に前記リソースの割当を解除する割当解除工程とを備えていることを要旨とするものである。

【0020】 また、本発明に係る他の分散ガーベッジコレクションのためのリソースの管理方法は、あるリース期間、あるリソースに対してアクセスすることをプロセスから要求する工程と、前記リソースに対する共有アクセスが許可されている間、承認されたリース期間を前記プロセスから受信する工程と、前記承認されたリース期間が終わりかけているが、前記リソースへのアクセスが完了していないという決定に基づいて、新たなリース期間の要求を前記プロセスに対して送信する工程とを備えていることを要旨とするものである。

【0021】 本発明に係る分散ガーベッジコレクションのための装置は、リソースを参照し、かつ要求されたりース期間を指定するプロセスからの要求を受信するように構成された受信モジュールと、承認されたリース期間、前記リソースへの共有アクセスを許可するように構成されたリソースアロケーターと、前記プロセスに前記承認されたリース期間を通知するように構成された通知モジュールと、前記承認されたリース期間が終了した時に前記リソースの割当を解除するように構成されたリソースディアロケーターとを備えていることを要旨とするものである。

【0022】 また、本発明に係る他の分散ガーベッジコレクションのための装置は、あるリース期間、あるリソースに対してアクセスすることをプロセスから要求するように構成された要求モジュールと、前記リソースに対する共有アクセスが許可されている間、承認されたリース期間を前記プロセスから受信するように構成された受信モジュールと前記承認されたリース期間が終わりかけているが、前記リソースへのアクセスが完了していないという決定に基づいて、新たなリース期間についての他の要求を、前記プロセスに対して送信するように構成された第2送信モジュールとを備えていることを要旨とするものである。

【0023】 本発明に係る分散ガーベッジコレクションのためのコンピュータプログラム生産物は、リソースを50 管理するために具体化されたコンピュータで読み取り可

能なコードを有するコンピュータ使用可能媒体を備えたコンピュータプログラム生産物において、前記コンピュータ使用可能媒体は、リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信するように構成された受信モジュールと、承認されたリース期間、前記リソースへの共有アクセスを許可するように構成されたリソースアロケーターと、前記承認されたリース期間を通知するように構成された通知モジュールと、前記承認されたリース期間が終了した時に前記リソースの割当を解除するように構成されたリソースディアロケーターとを備えていることを要旨とするものである。

【0024】また、本発明に係る他の分散ガーベッジコレクションのためのコンピュータプログラム生産物は、リソースを管理するために具体化されたコンピュータで読み取り可能なコードを有するコンピュータ使用可能媒体を備えたコンピュータプログラム生産物において、前記コンピュータ使用可能媒体は、あるリース期間、あるリソースに対してアクセスすることをプロセスから要求するように構成された要求モジュールと、前記プロセスが前記リソースに対する共有アクセスを許可している間、承認されたリース期間を前記プロセスから受信するように構成された受信モジュールと、前記承認されたリース期間が終わりかけているという決定に基づいて、新たなリース期間についての他の要求を、前記プロセスに対して送信するように構成された送信モジュールとを備えていることを要旨とするものである。

【0025】本発明に係る分散ガーベッジコレクションのためのコンピュータシステムは、コンピュータプロセッサーと、前記コンピュータプロセッサーと有効に接続されているメモリーと、前記メモリーにより前記コンピュータプロセッサー中で実行するコンピュータプロセスを備え、前記コンピュータプロセスは、リソースを参照し、かつ要求されたリース期間を指定するプロセスからの要求を受信するように構成された受信モジュールと、承認されたリース期間、前記リソースに対する共有アクセスを許可するように構成されたリソースアロケーターと、前記承認されたリース期間を通知するように構成された通知モジュールと、前記承認されたリース期間が終了した時に前記リソースの割当を解除するように構成されたリソースディアロケーターとを備えていることを要旨とするものである。

【0026】また、本発明に係る他の分散ガーベッジコレクションのためのコンピュータシステムは、コンピュータプロセッサーと、前記コンピュータプロセッサーと有効に接続されているメモリーと、前記メモリーにより前記コンピュータプロセッサー中で実行するコンピュータプロセスを備え、前記コンピュータプロセスは、あるリース期間、あるリソースに対してアクセスすることをプロセスから要求するように構成された要求モ

ジュールと、前記プロセスが前記リソースに対する共有アクセスを許可している間、承認されたリース期間を前記プロセスから受信するように構成された受信モジュールと、前記承認されたリース期間が終わりかけているという決定に基づいて、新たなリース期間についての他の要求を、前記プロセスに対して送信するように構成された送信モジュールとを備えていることを要旨とするものである。

【0027】さらに、本発明に係る分散ガーベッジコレクションのためのプラットフォーム間に分散したリソースの割当及び割当解除方法は、複数のプラットフォームを有する分散処理システムにおいて、サーバープラットフォームで実行される以下の工程、すなわち、(a) システムリソースに対するリファレンス及び要求されたリース期間を含むリクエスター プラットフォームからの要求であって、前記対応するリクエスター プラットフォームが前記要求中において前記要求された期間、前記リファレンスされたシステムリソースへのアクセスをシークすることを示している要求を受信する工程と、(b)

(i) 前記要求された各リース期間に応答して設定される、対応する承認期間であって、要求に関連する前記リファレンスされた各システムリソースが前記プラットフォームによってアクセスされることが可能となる期間を示している承認期間を設定すること、及び (ii)

対応するリファレンスカウンターであって、要求の受信によって決定される前記対応するシステムリソースがアクセスしている前記プラットフォームの数を示しているリファレンスカウンターを増加させること、により前記リファレンスされた各システムリソースを割り当てる工程と、(c) 関連する要求に応答して、関連する要求中でリファレンスされた前記リソースのための前記承認期間を指定する前記リクエスター プラットフォームに対する応答を送信する工程と、(d) (i) リファレンスされた各システムリソースのための前記承認期間が終了した時、又は (ii) 前記対応するリファレンスカウンターが、いかなるリクエスター プラットフォームも、現在、前記リファレンスされた各システムリソースに対するアクセスをシークしていないことを示す状態である時に、前記リファレンスされた各システムリソースの割当を解除するためのガーベッジコレクションプロセスを起動する工程とを備えたことを要旨とするものである。

【0028】上記構成を有する本発明に係るリソースの管理方法によれば、リファレンスの保全性は、ある一定の期間、すなわち、分散システム中のパーティ、例えば、リソースに対するリファレンスを保持しているアプリケーション及びそのリソースを管理しているガーベッジコレクションシステムが、そのリソースとそのリソースに対するリファレンスとが保証されることについて合意している期間、リソースをリースすることによって、

大きな記憶漏れを伴うことなく保証される。

【0029】そのリース期間が終了したときには、そのリソースに対するリファレンスが継続するという保証は、消滅し、ガーベッジコレクションシステムにそのリソースをリクレームすることを許可する。なぜならば、そのリソースに対するリファレンスを保持しているアプリケーションと、そのリソースを管理しているガーベッジコレクションシステムは、有限の保証されたリース期間について合意しており、双方とも、いつそのリースが終了するか、すなわちいつその保証が終了するかを知ることができるからである。これにより、リファレンスのリース期間中はリファレンスの保全性が保証され、ネットワークエラーに起因して発生するリソースの開放失敗を回避することが可能となる。

【0030】

【発明の実施の形態】以下に、添付した図面の記載に基づいて、本発明の実施の形態を詳細に説明する。可能な限り、同一又は類似の部分について言及する図面及び以下の記述全体を通して、同一の参照番号を使用する。

【0031】本発明は、ありふれた分散処理システムの構成によって組織されたコンピュータを用いて実行することができる。しかしながら、本発明を実行するための構成と手順は、ありふれたものではない。なぜならば、それらは、リファレンスの保全性を保証し、記憶漏れを排除する分散ガーベッジコレクション機構を与えるものだからである。

【0032】(A. 総括) 分散処理システム中にある各コンピュータに配置されている方法呼出要素(以下、「M I要素」という)が、本発明に係る分散ガーベッジコレクション機構を実行する。M I要素は、好ましくは、プログラム言語JAVA(登録商標)によって書かれた多くのソフトウェアモジュールからなっている。

【0033】一般に、分散処理システム中のアプリケーションが、分散リソースに対するリファレンスを、ネームの索引あるいは他の方法によって、ある他のコールに対するリターン値として取得し、そのリソースへのアクセスをシークする時はいつでも、そのアプリケーションは、そのリソースに対し、あるいはそのリソースを管理するM I要素に対してコールを行う。

【0034】マネージングM I要素と呼ばれるこのM I要素は、そのリソースに対して突出しているリファレンス数のトラックをキープする。あるリソースに対するリファレンスの数がゼロの時は、マネージングM I要素は、そのリソースをリクレームすることができる。あるリソースに対するリファレンス数のカウントは、一般に、「リファレンスカウント」と呼ばれており、リファレンスカウントを増加させるコールは、「ダーティーコール」と称される。

【0035】アプリケーションが、もはや分散リソースを要求しない時は、アプリケーションは、そのリソース

に対しあるいはマネージングM I要素に対して、異なるコールを送信する。このコールの受信に基づき、マネージングM I要素は、そのリソースのリファレンスカウントを減少させる。このリファレンスを落とすためのコールは、「クリーンコール」と称される。

【0036】本発明に係る実施の形態によれば、ダーティーコールには、リース期間と呼ばれる、そのリソースに対するリファレンスのための要求された時間間隔を含めることができる。ダーティーコールの受信に基づき、

10 マネージングM I要素は、リースが承認された期間を含んでいるリターンコールを送信する。すなわち、マネージングM I要素は、突出したリファレンスの数と同様に、それらのリファレンスのためのリース期間もトラックする。その結果、あるリソースに対するリファレンスカウントがゼロになった時、あるいはそのリソースのためのリース期間が終了した時には、マネージングM I要素は、そのリソースをリクレームすることができる。

【0037】(B. 手順) M I要素中のアプリケーションコールプロセッサーは、図1に図示されたアプリケーションコール手順100のステップを実行する。マネージングM I要素中のサーバーコールプロセッサーは、それぞれ図2~4に図示される、手順200、300及び400のステップを実行する。マネージングM I要素のガーベッジコレクターは、サーバーコールプロセッサーからの命令に従って、それ以前にリファレンスと結合していたリソースをリクレームするために通常の手順を実行する。従って、ガーベッジコレクターの通常の手順については、説明を省略する。

【0038】(1. アプリケーションコールプロセッサー) 図1は、分散処理システム中に配置されている同一又は他のM I要素によって管理されるリソースに対するリファレンスのためのアプリケーションの要求を取り扱うために、M I要素のアプリケーションプロセッサーが使用する手順100のフロー図を示したものである。

【0039】アプリケーションがあるリソースに対するリファレンスを取得した後、アプリケーションコールプロセッサーは、リソースのリファレンス及び要求されたリース期間を含むダーティーコールを、そのリソースのためのマネージングM I要素に対して送信する(ステップ110)。ダーティーコールは、リソースそれ自体に対してあるいはマネージングM I要素に対して向けられる。

【0040】次いで、アプリケーションコールプロセッサーは、マネージングM I要素からのリターンコールを待ち、そして受信する(ステップ120)。そのリターンコールには、ダーティーコールのリファレンスがそのリソースと結合するであろうということをマネージメントM I要素が保証する期間、すなわち、承認されたリース期間が含まれている。換言すれば、マネージングM I要素は、その承認期間、ダーティーコールのリファレン

スに対応するリソースを収集しないことに同意する。もし、マネージングM I要素が承認期間を与えないか、あるいはリースの要求を拒絶するならば、アプリケーションコールプロセッサーは、承認期間を受信するまで、別のダーティーコールを送信しなければならない。

【0041】アプリケーションコールプロセッサーは、アプリケーションによるそのリファレンスの使用を監視し、そのアプリケーションが、アプリケーションコールプロセッサーに対し、そのリファレンスがもはや要求されないとということを明確に知らせた時、又は、そのアプリケーションプロセッサーが、自分自身でその決定を下した時のいずれかの時（ステップ130）には、アプリケーションコールプロセッサーは、マネージングM I要素に対してクリーンコールを送信する（ステップ140）。ダーティーコールに用いられた方法と同様の方法により、クリーンコールは、リファレンスされたリソースに向けられ、マネージングM I要素は、クリーンコールを処理する。続いて、アプリケーションコールプロセッサーは、アプリケーションによって使用されているリファレンスのリストからそのリファレンスを削除する（ステップ150）。

【0042】もし、アプリケーションが、いまだリファレンスをやめていない（ステップ130）が、アプリケーションコールプロセッサーが、そのリファレンスのための承認期間が終了しそうであると判断する場合（ステップ160）には、次に、アプリケーションコールプロセッサーは、そのリソースに対するリファレンスが、アプリケーションに代わってマネージングM I要素によって保持されているということを保証するために、ステップ110とステップ120を繰り返す。

【0043】（2. サーバーコールプロセッサー）M I要素のサーバーコールプロセッサーは、3つの主要な手順を実行する。すなわち、（1）ダーティーコールを処理すること、（2）入ってくるクリーンコールを処理すること、及び（3）適切なときに、リソースをリクレームするためのガーベッジコレクションサイクルを起動させること、である。

【0044】（i. ダーティーコール）図2は、M Iソフトウェア要素が管理するリファレンスされたリソースに対する要求、すなわちダーティーコールを処理するために、M I要素のサーバーコールプロセッサーが使用する手順200のフロー図である。これらの要求は、分散処理システム中にあるM I要素のアプリケーションコールプロセッサーからくるものであり、そのアプリケーションコールプロセッサーには、要求を処理するサーバーコールプロセッサーと同じM I要素中のアプリケーションコールプロセッサーも含まれている。

【0045】最初に、サーバーコールプロセッサーは、ダーティーコールを受信する（ステップ210）。次いで、サーバーコールプロセッサーは、受入可能な承認期

間を決定する（ステップ220）。承認期間は、要求されたリース期間、あるいは他の期間と同一でもよい。サーバーコールプロセッサーは、要求されたリソースの量及び同一のリソースに対してそれ以前に承認された他の多くの承認期間を含む多数の条件に基づいて適切な承認期間を決定する。

【0046】リソースがダーティーコールのリファレンスにより未だ割り当てられていないとサーバーコールプロセッサーが判断した時（ステップ230）には、サーバーコールプロセッサーは、要求されたリソースを割り当てる（ステップ240）。

【0047】次いで、サーバーコールプロセッサーは、ダーティーコールのリファレンスに対応するリファレンスカウントを増加させ（ステップ250）、リファレンス・リソース結合のための受入可能な承認期間を設定し（ステップ260）、アプリケーションプロセッサーに対し承認期間と共にリターンコールを送信する（ステップ270）。このようにして、サーバーコールプロセッサーは、その制御下にあるリソースに対するリファレンスに注意しながら、入ってくるダーティーコールを制御する。

【0048】アプリケーションは、現在のリースが終了する前に、拡張要求を伴うダーティーコールを送信することによって、リースを延長することができる。手順200において示したように、リースを拡張するための要求は、リースのための最初の要求と全く同様に扱われる。拡張は、単に、リファレンスカウントがゼロにならない限り、ある付加的な期間、リソースはリクレームされないということを意味するにすぎない。

【0049】（i i. クリーンコール）M I要素のサーバーコールプロセッサーは、また、アプリケーションコールプロセッサーから入ってくるクリーンコールも処理する。分散処理システム中のアプリケーションがもはやリソースに対するリファレンスを要求しない時は、アプリケーションは、そのリソースが再使用のためにリクレームされるように、そのリファレンスのためのリソースを管理するM I要素に知らせる。図3は、M I要素のサーバーコールプロセッサーがクリーンコールを処理するために使用するステップを有する手順300のフロー図を示している。

【0050】サーバーコールプロセッサーが、M I要素が管理するリソースに対するリファレンスと共にクリーンコールを受信した時（ステップ310）は、サーバーコールプロセッサーは、対応するリファレンスカウントを減少させる（ステップ320）。クリーンコールは、そのリソースを監視し、かつそのコールを処理するための手順300を実行するとサーバーコールプロセッサーと共に、そのリソースに送信される。続いて、サーバーコールプロセッサーは、受信を承認するために、クリーンコールを送信しているM I要素に対してリターンコー

ルを送信する（ステップ330）。本発明に係る実施の形態によれば、リファレンスを落とすためのクリーンコールは拒絶されないが、承認されなければならない。

【0051】(iii. ガーベッジコレクション) サーバーコールプロセッサーは、また、もはやリソースに対してリファレンスがなされないか、あるいはそのリソースのための合意されたリース期間が終了したかのいずれかを決定したそのリソースをリクレームするために、ガーベッジコレクションサイクルを起動させる。図4に示した手順400は、サーバーコールプロセッサーがガーベッジコレクションサイクルを起動するために使用するステップのフロー図を示している。

【0052】サーバーコールプロセッサーは、リファレンスカウント及び承認されたリース期間を監視し、M I要素によって管理されているあるリソースについてリファレンスカウントがゼロであるか否か、あるいはあるリファレンスのための承認期間が終了したか否かを決定する（ステップ410）。いずれかの条件が存在するときは、サーバーコールプロセッサーは、そのリソースのガーベッジコレクションを起動する（ステップ420）。そうでない場合は、サーバーコールプロセッサーは、リファレンスカウントと承認されたリース期間の監視を続行する。

【0053】(C. コールフロー) 図5は、分散処理システム内にあるM I要素間のコールの流れを模式的に表した図である。マネージングM I要素525は、リソース530に対するリファレンスを監視することによって、リソース530を管理する（ガーベッジコレクト505参照）。マネージングM I要素525は、リソースを管理するので、マネージングM I要素525のサーバーコールプロセッサーは、ここに記載されたコールフローの処理を実行する。

【0054】図5は、また、アプリケーション510及び540が、それぞれ、対応するM I要素515及び545を有していることを示している。各アプリケーション510及び540は、それぞれ、リソース530の内の一つに対するリファレンスを取得し、リファレンスが対応するリソースと結合するように、リソース530の内の一つに対するアクセスを取得するためにシークする。

【0055】アクセスを取得するために、アプリケーション510及び540は、M I要素525に対して、それぞれ、ダーティーコール551及び571を送信するために、それぞれ、対応するM I要素515及び545を呼び出す。M I要素515及び545は、マネージングM I要素525のような他のM I要素によって管理されるリソース530に対してアクセスするためのアプリケーションの要求を処理するので、M I要素515及び545のアプリケーションコールプロセッサーは、ここに記載されたコールフローの処理を実行する。

【0056】ダーティーコール551及び571に応答して、マネージングM I要素525は、各M I要素515及び545に対して、それぞれ、リターンコール552及び572を送信する。ダーティーコールには、ダーティーコール551及び571のリファレンスのための承認されたリース期間を含んでいる。

【0057】同様に、図5は、また、マネージングM I要素525に対して、それぞれ、クリーンコール561及び581を送信しているM I要素515及び545を示している。クリーンコール561及び581は、アプリケーション510及び540が、それぞれ、クリーンコール561及び581で指定されるリソースに対するアクセスをもはや要求しないということをマネージングM I要素525に知らせる。マネージングM I要素525は、クリーンコール561及び581に対して、それぞれ、リターンコール562及び582で応答する。リターンコール562及び582は、それが、受信したクリーンコール561及び581に対するM I要素525からの単なる承認であるという点において、リターンコール552及び572とは異なる。

【0058】アプリケーション510及び540の双方とも、同一リソースに対するアクセスを要求するかもしれない。例えば、アプリケーション540がそれ以前に「リソース（1）」へのアクセスが承認されていた一方で、アプリケーション510が、「リソース（1）」に対するアクセスを要求することもある。M I要素525は、合意されたリース期間、アプリケーション510及び540の双方がそのリソースを使用できるようにすることによってこのような状況を処理する。すなわち、M I要素525は、アプリケーション510及び540の双方とも、そのリソースに対するリファレンスを落とすか、あるいは最新の合意された期間が終了したかのいずれかの事象が最初に生ずるまで、「リソース（1）」をリクレームするためにガーベッジコレクションサイクルを起動させない。

【0059】1より多いアプリケーションに対し、同一のリソースに対して同時にアクセスすることを許可することによって、本発明は、また、アプリケーションが、リソースに対するリファレンスを落とすように、マネージングM I要素に対してクリーンコールを送信した後に、そのアプリケーションがそのリソースにアクセスすることを許可する。これは、そのリソースが他のアプリケーションによっていまなおリファレンスされているか、あるいはそのリファレンスのリースが未だに終了しておらず、それによりマネージングM I要素525が未だにそのリソースをリクレームしていないために生ずる。しかしながら、そのリソースは、有限の期間が経過した後、すなわち、いかなるアプリケーションもリースしていない時、あるいは最新のリースが終了した時に、リクレームされることになる。

【0060】(D. M I要素) 図6は、本発明の実施の形態に係るM I要素6 0 0のモジュールのブロック図を示している。M I要素6 0 0には、監視されている各リファレンスのためのリファレンス要素6 0 5、アプリケーションコールプロセッサー6 4 0、サーバーコールプロセッサー6 5 0及びガーベッジコレクター6 6 0を含めることができる。

【0061】リファレンス要素6 0 5は、好ましくは、リファレンスデータポーション6 1 0に備えられた表又はこれに相当する構造、リファレンスカウント6 2 0及び承認期間レジスター6 3 0で構成する。M I要素6 0 0は、対応するリソースをリクレームするためにガーベッジコレクター6 6 0をいつ起動させるかを決定するために、対応するリファレンスデータポーション6 1 0で指定される各リファレンスのための、リファレンスカウント6 2 0と承認期間6 3 0を使用する。

【0062】アプリケーションコールプロセッサー6 4 0は、図1に示す手順1 0 0のステップを実行するソフトウェアモジュールである。サーバーコールプロセッサー6 5 0は、図2～4に示す手順2 0 0、3 0 0及び4 0 0のステップを実行するソフトウェアモジュールである。ガーベッジコレクター6 6 0は、上述したように、サーバーコールプロセッサー6 5 0からの指令に応答してリソースをリクレームするソフトウェアモジュールである。

【0063】(E. 分散処理システム) 図7は、本発明を実施するために用いられる分散処理システム5 0を図示したものである。図7において、分散処理システム5 0は、ネットワーク雲5 5によって表現されるネットワーク配列中に接続された3つの独立で、かつ異質なプラットフォーム7 0 0、8 0 0及び9 0 0を含んでいる。図7中で、雲5 5によって表現されるネットワーク配列の構成及びプロトコルは、プラットフォーム7 0 0、8 0 0及び9 0 0間の情報伝達を許すものである限り、重要なことではない。

【0064】加えて、ちょうど3つのプラットフォームを使用していることは、単なる例示にすぎないものであり、本発明が、非常に多くのプラットフォームを使用することを制限するものではない。さらに、特定のネットワーク構造は、本発明にとって重要なことではない。例えば、本発明に従って使用することができる他のネットワーク構造では、他の全てのプラットフォームが接続されるネットワークコントローラーとして、1つのプラットフォームを使用する。

【0065】分散処理システム5 0の実施の形態において、プラットフォーム7 0 0、8 0 0及び9 0 0は、それぞれ、プロセッサー7 1 0、8 1 0及び9 1 0、並びにメモリー7 5 0、8 5 0及び9 5 0を含んでいる。各プロセッサー7 1 0、8 1 0及び9 1 0には、それぞれ、アプリケーション7 2 0、8 2 0及び9 2 0、オペ

ーティングシステム7 4 0、8 4 0及び9 4 0、並びにM I要素7 3 0、8 3 0及び9 3 0が含まれている。

【0066】アプリケーション7 2 0、8 2 0及び9 2 0は、本発明を実行するために以前に書かれかつ修正されたプログラム、あるいは本発明によって提供されるサービスを利用するため特に書かれたプログラムのいずれでもよい。アプリケーション7 2 0、8 2 0及び9 2 0は、本発明に従って実行されるオペレーションを呼び出す。

【0067】M I要素7 3 0、8 3 0及び9 3 0は、図6を参照しながら前述したM I要素6 0 0に対応している。

【0068】オペレーティングシステム7 4 0、8 4 0及び9 4 0は、それぞれ、対応するプロセッサー7 1 0、8 1 0及び9 1 0と連結された標準的なオペレーティングシステムである。プラットフォーム7 0 0、8 0 0及び9 0 0は、異質なものとすることができます。例えば、プラットフォーム7 0 0は、プロセッサー7 1 0として、サン・マイクロシステムズ社製ウルトラスパークマイクロプロセッサーを有し、オペレーティングシステム7 4 0としてソラリスを使用する。

【0069】プラットフォーム8 0 0は、プロセッサー8 1 0として、シリコングラフィクス社製MIPSマイクロプロセッサーを有し、オペレーティングシステム8 4 0としてユニックスを使用する。最後に、プラットフォーム9 0 0は、プロセッサー9 1 0としてインテル社製ペンティアムマイクロプロセッサーを有し、オペレーティングシステム9 4 0としてマイクロソフトウンドウズ9 5を使用する。本発明は、これに限定されるものではなく、さらに同質のプラットフォームを収容することもできる。

【0070】サン、サン・マイクロシステムズ、ソラリス、ジャバ及びサンロゴは、米国及び他の国におけるサン・マイクロシステムズ社の商標又は登録商標である。ウルトラスパーク及び他の全てのスパーク商標は、許諾に基づいて使用されており、米国及び他の国におけるスパークインターナショナル社の商標である。スパーク商標を付した製品は、サン・マイクロシステムズ社によって開発された構造を基礎に置いている。

【0071】メモリー7 5 0、8 5 0及び9 5 0は、関連するプラットフォームのための一般的な記憶のようないくつかの機能を与える。他の機能は、各プロセッサー7 1 0、8 1 0及び9 1 0による実行の前に、アプリケーション7 2 0、8 2 0及び9 2 0、M I要素7 3 0、8 3 0及び9 3 0、並びにオペレーティングシステム7 4 0、8 4 0及び9 4 0を記憶することである。加えて、メモリー7 5 0、8 5 0及び9 5 0の一部は、ネットワーク5 0中の全てのプラットフォーム7 0 0、8 0 0及び9 0 0が使用可能な共有メモリーを構成してもよい。

【0072】(E. M I サービス) 本発明は、クライアント／サーバーモデルを用いて実行してもよい。クライアントは、ダーティーコールやクリーンコールのような要求を発生させ、サーバーは、その要求に対して応答する。

【0073】図7に示す各M I要素730、830及び930は、クライアント要素とサーバー要素の双方を含んでいることが望ましい。図8は、クライアントプラットフォーム1000及びサーバープラットフォーム1100のブロック図を示しており、図7中のプラットフォーム700、800及び900のいずれか二つに当たる。

【0074】プラットフォーム1000及び1100は、それぞれ、メモリー1050及び1150、並びにプロセッサー1010及び1110を有している。プラットフォーム1000及び1100中の要素は、図7を参照しながら前述した同様の要素と同じ方法で動作する。この例では、プロセッサー1010は、クライアントアプリケーション1020を実行し、プロセッサー1110は、サーバーアプリケーション1120を実行する。プロセッサー1010及び1110は、また、それぞれ、オペレーティングシステム1040及び1140、並びにM I要素1030及び1130を実行する。

【0075】M I要素1030及び1130は、それぞれ、サーバーコールプロセッサー1031及び1131、アプリケーションコールプロセッサー1032及び1132、並びにガーベッジコレクター1033及び1133を有している。各M I要素1030及び1130は、また、各M I要素1030及び1130が監視する各リファレンスのための、リファレンスデータポーション1034及び1134、リファレンスカウント1035及び1135、並びに承認期間レジスター1036及び1136を含むリファレンス要素を有している。

【0076】アプリケーションコールプロセッサー1032及び1132は、クライアントサービスに相当し、それぞれ、サーバーサービスに相当するサーバーコールプロセッサー1031及び1131と相互に情報伝達を行う。プラットフォーム1000及び1100は、サーバーコールプロセッサー、アプリケーションコールプロセッサー、ガーベッジコレクター及びリファレンス要素を有しているので、いずれのプラットフォームも、クライアントあるいはサーバーとして機能することができる。

【0077】しかしながら、後述する議論のために、プラットフォーム1000は、クライアントプラットフォームを示し、プラットフォーム1100は、サーバープラットフォームを示すものとする。この例では、クライアントアプリケーション1020は、分散リソースに対してリファレンスを取得し、サーバープラットフォーム1100のM I要素1130によって管理されるリソ

スに対してダーティーコールを送るために、M I要素1030を使用する。

【0078】加えて、サーバープラットフォーム1100は、サーバーアプリケーション1120を実行する。サーバーアプリケーション1120は、また、ダーティーコールを送信するためにM I要素1130を使用する。そして、ダーティーコールは、そのダーティーコールのリソースがM I要素1130によって管理されている時は、M I要素1130によって処理される。あるいは、サーバーアプリケーション1120は、M I要素1030で管理されているリソースに対してダーティーコールを送信するためにM I要素1130を使用する。

【0079】従って、クライアントプラットフォーム1000中のM I要素1030のためのサーバーコールプロセッサー1031、ガーベッジコレクター1033及びリファレンスカウント1035は、使用されていないので、図8中においては陰影が付されている。同様に、サーバープラットフォーム1100中のM I要素1130のアプリケーションコールプロセッサー1132もまた、休止状態であるので、陰影が付されている。

【0080】クライアントアプリケーション1020があるリソースに対応するリファレンスを取得した時には、アプリケーションコールプロセッサー1032は、ダーティーコールを送信し、それをサーバーコールプロセッサー1131が受信する。ダーティーコールには、要求されたリース期間が含まれている。サーバーコールプロセッサー1131は、ダーティーコール中のリファレンスのためにリファレンスカウント1135を増加させ、承認期間を決定する。これに応じて、サーバーコールプロセッサー1131は、アプリケーションコールプロセッサー1030に対して承認期間と共にリターンコールを送信する。アプリケーションコールプロセッサー1032は、記録されている承認期間1035を更新するため、及びダーティーコールのリファレンスに対応するリソースをいつリクレームするかを決定するためにその承認期間を使用する。

【0081】サーバーコールプロセッサー1131は、また、それが管理するリソースのためのリファレンスに対応するリファレンスカウントと承認期間を監視する。40 そのリファレンスカウント1135の内の一つがゼロの時、又はそのリファレンスのための承認期間1135が終了した時のいずれかの事象が最初に生じた場合は、いつでも、サーバーコールプロセッサー1131は、リファレンスカウントがゼロ又は承認期間が終了したリファレンスに対応するリソースをリクレームするためにガーベッジコレクター1133を起動してよい。

【0082】本発明の実施の形態に係るリース・リファレンス機構は、プロトコルに含まれるプラットフォーム1000及び1100上のクロックが同期していること50 を要求していない。その機構は、それらが比較しうる増

加期間を有することを、単に要求するだけである。リースは、特定の時刻に終了するのではなく、むしろ一定の時間間隔が経過した後に終了する。時間間隔がほぼ一致している限り、プラットフォーム1000及び1100は、承認されたリース期間について、およその合意が得られる。さらに、リースのためのタイミングは、コンピューターの見地からすれば、かなり長いので、クロックレートのわずかな差異は、ほとんど、あるいは全く影響がない。

【0083】ダーティーコールの転送時間は、プロトコルの影響を受ける。もし、M1要素1030がリファレンスに対するリースを保持し、更新の要求をするために、リースが終了する直前までウェイト状態であるならば、リースは、M1要素1130がその要求を受信する前に終了するかもしれない。もし、そうであるならば、M1要素1130は、更新要求を受信する前にリソースをリクレームするかもしれない。すなわち、ダーティーコールを送るときには、送り手側は、そのリソースのためのリース期間が終了する前に更新のダーティーコールがなされるように、ダーティーコールのリソースを処理しているプラットフォームへの転送時間を考慮して、要求されたリース期間に対して、時間因子を加えるべきである。

【0084】(F. 結論) 本発明によれば、分散ガーベッジコレクション機構は、承認されたリース期間が終了した時に、リソースに対するリファレンスが終了するよう、分散処理システム中のリソースに対するリファレンスに対応する承認されたリース期間を与えることによって、リファレンスの保全性を保証し、記憶漏れを排除する。その後に、そのリソースを収集するようにしてもよい。また、リソースに対するリファレンスに付与されたカウンターを参照して、リソースがもはや分散処理システム中のプロセスによってリファレンスされない時に、リソースを収集するようにしてもよい。

【0085】本発明は、上気した実施の形態に何ら限定されるものではなく、本発明の要旨を逸脱しない範囲で種々の改変が可能である。例えば、前述の実施の形態には、ソフトウェアが含まれているが、本発明は、ハードウェアとソフトウェアの組み合わせ、あるいはハードウェアのみで実施することができる。

【0086】

【発明の効果】本発明によれば、リファレンスの保全性は、ある一定の期間、すなわち、分散システム中のバー

ティ、例えば、リソースに対するリファレンスを保持しているアプリケーション及びそのリソースを管理しているガーベッジコレクションシステムが、そのリソースとそのリソースに対するリファレンスとが保証されることについて同意している期間、リソースをリースすることによって、大きな記憶漏れを伴うことなく保証される。

【0087】そのリース期間が終了したときには、そのリソースに対するリファレンスが継続するという保証は、消滅し、ガーベッジコレクションシステムにそのリソースをリクレームすることを許可する。なぜならば、そのリソースに対するリファレンスを保持しているアプリケーションと、そのリソースを管理しているガーベッジコレクションシステムは、有限の保証されたリース期間について合意しており、双方とも、いつそのリースが終了するか、すなわちいつその保証が終了するかを知ることができるからである。これにより、リファレンスのリース期間中はリファレンスの保全性が保証され、ネットワークエラーに起因して発生するリソースの開放失敗を回避することが可能となるという効果がある。

【図面の簡単な説明】

【図1】本発明の実施の形態に係る、アプリケーションコールプロセッサーによって実行されるステップを示すフロー図である。

【図2】本発明の実施の形態に係る、ダーティーコールを処理するためのサーバーコールプロセッサーによって実行されるステップを示すフロー図である。

【図3】本発明の実施の形態に係る、クリーンコールを処理するためのサーバーコールプロセッサーによって実行されるステップを示すフロー図である。

【図4】本発明の実施の形態に係る、ガーベッジコレクションプロセスを起動するためのサーバーコールプロセッサーによって実行されるステップを示すフロー図である。

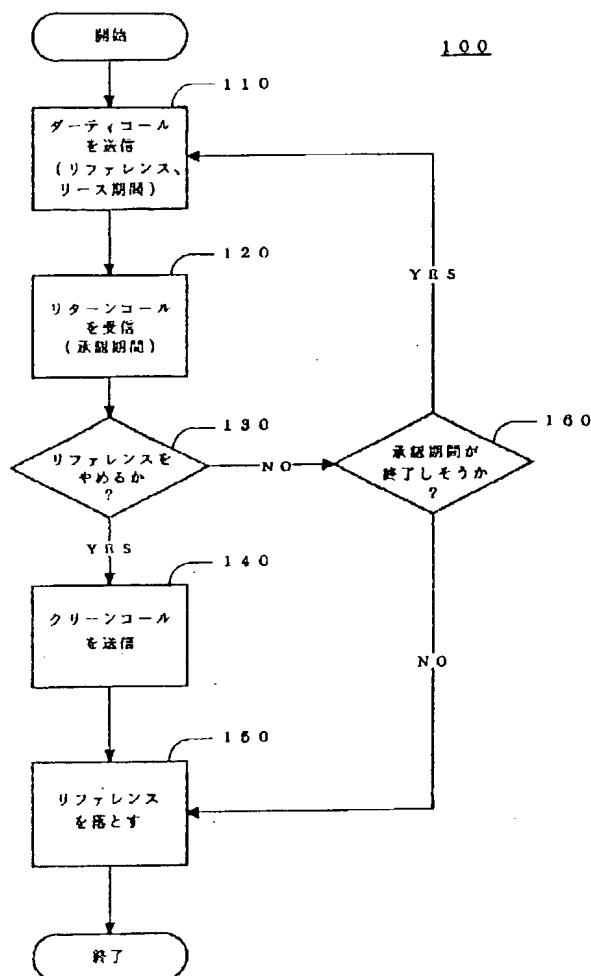
【図5】分散処理システム内のコールの好ましい流れを示す図である。

【図6】本発明に係る方法呼出作業を遂行する構成要素のブロック図である。

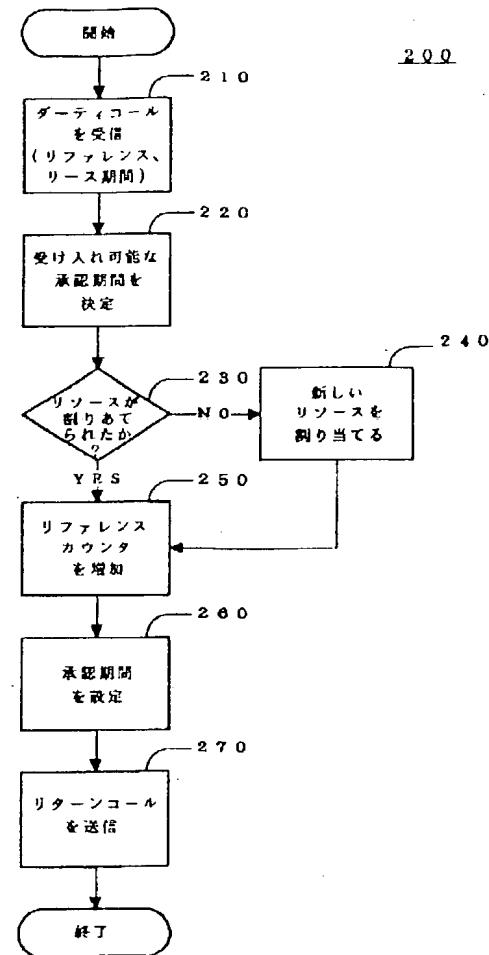
【図7】本発明の実施の際に用いられる分散処理システムを示す図である。

【図8】本発明の実施の形態に係る、分散処理システムのプラットフォームにおける個々のソフトウェアの構成要素を示す図である。

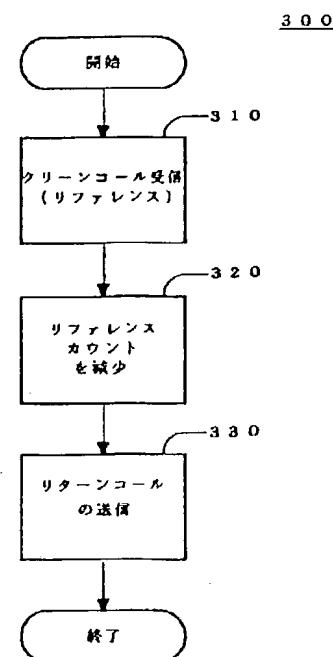
【図1】



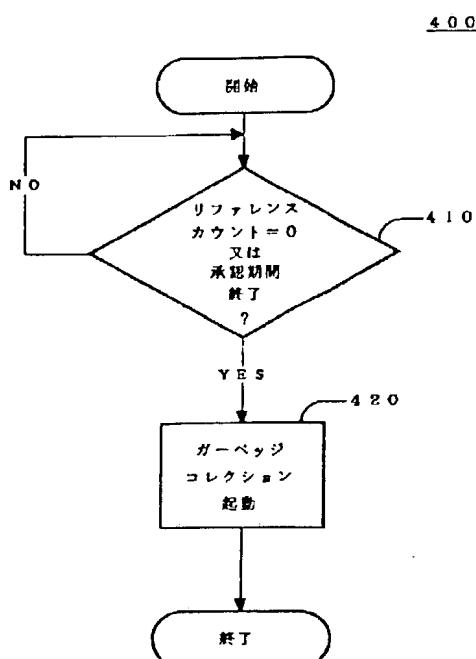
【図2】



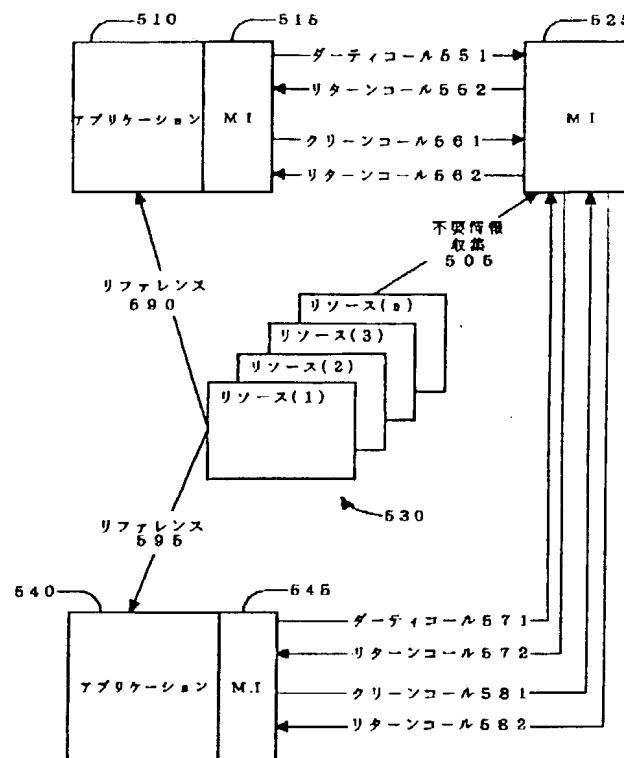
【図3】



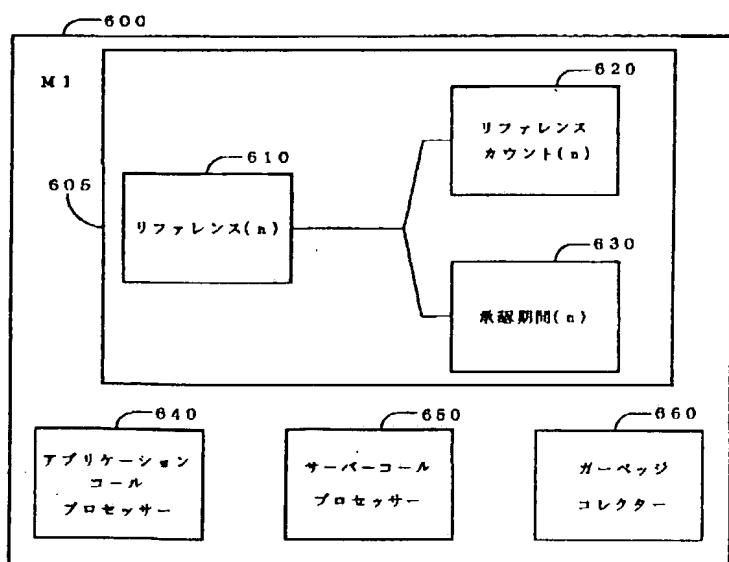
【図4】



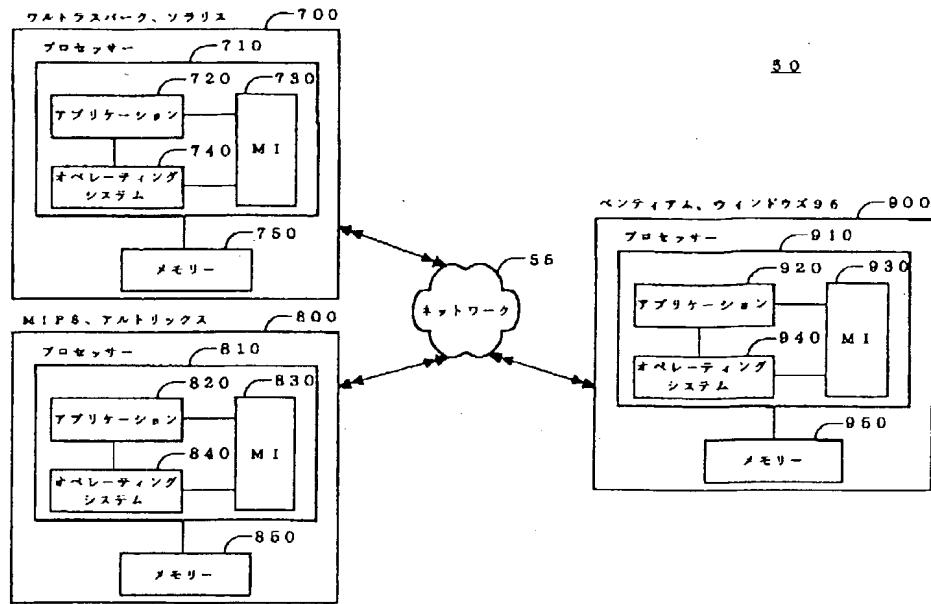
【図5】



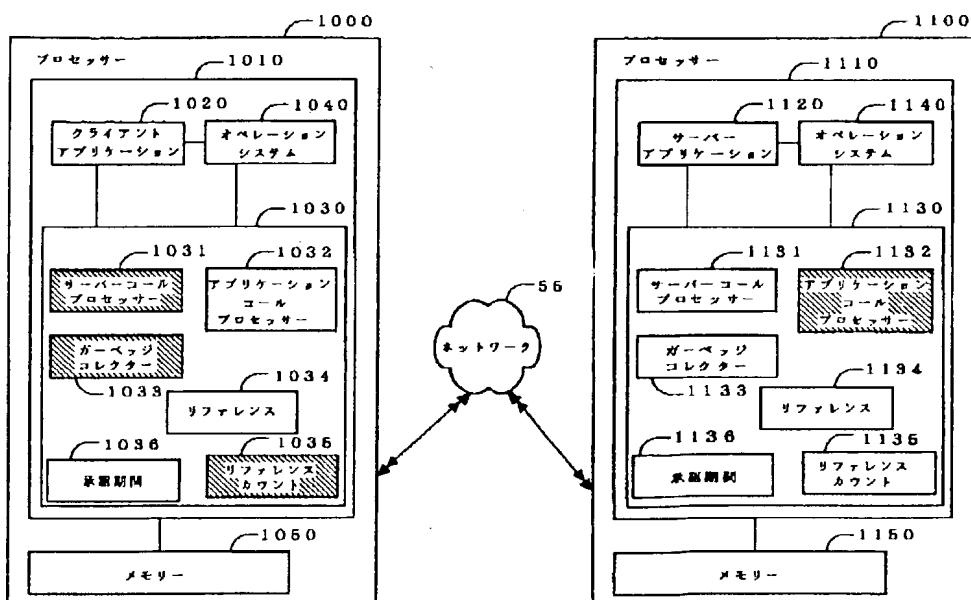
【図6】



【図7】



【図8】



フロントページの続き

(72)発明者 ウォールドー ジェームス エイチ
アメリカ合衆国、01826 マサチューセッ
ツ州、ドラカット、ラビーロード155

(72)発明者 リッグス ロジャー
アメリカ合衆国、01826 マサチューセッ
ツ州、バーリントン、ブライアウッドレイ
ン4

(54) 【発明の名称】 分散ガーベッジコレクションのためのリソースの管理方法、装置、コンピュータプログラム生産物、コンピュータシステム並びにプラットフォーム間に分散したリソースの割当及び割当解除方法